



# Détection et étiquetage automatique d'image: Partie2

Par

Joël Brou Boni

P-SAT

Mr Elod Egyed-Zsigmond



## COMMENT UTILISER LE DETECTEUR AUTOMATIQUE DE VISAGE ?

Il s'agit tout d'abord d'avoir toutes les librairies indispensables à son utilisation.

Je me focalise dans ce document dans la description de DeepFace. Le rapport sur YOLOv3 a déjà été envoyé précédemment.

J'ai exécuté ce programme sur **Mac et Linux** sans problème, il devrait en être de même pour **Windows**.

Si cela ne marche pas sur cet OS, il s'agira certainement de dépendances liées aux fichiers de poids(h5) à télécharger sur google Drive et à placer dans un dossier précis. La procédure est décrite dans le fichier **deepfacev2.py** mais **n'est pas nécessaire dans la grande majorité des cas**.

Il faudra ensuite exécuter les lignes suivantes dans le répertoire **DeepFace**.

**pip install deepface**

**pip install opencv-python**

## UN RESEAU DE NEURONES EXPLOITANT DEEPFACE

- Vous devez avoir dans le répertoire **DeepFace** les fichiers **res10\_300x300\_ssd\_iter\_140000.caffemodel**, **deploy.prototxt.txt** et **deepfacev2.py**

Enfin, il y'a à disposition des images(**.jpg**) afin de tester l'efficacité de notre programme.

## FONCTIONNEMENT DES PROGRAMMES

### YOLOV3

Le fonctionnement est légèrement différent comparé à la 1<sup>ère</sup> partie

Un fichier **requirements.txt** est présent dans le répertoire **Image recognition** et récapitule toutes les dépendances requises.

Il faudra ensuite exécuter la ligne suivante dans le répertoire **Image recognition**.

**pip install -r requirements.txt**

- Le fichier de poids **YOLOV3** permettant l'étiquetage des objets étant trop lourd pour être transmis en zip, il est possible de les télécharger à cette adresse <https://pjreddie.com/media/files/yolov3.weights>.
- Une fois ce fichier **yolov3.weights** téléchargé et placé dans le répertoire courant, vous devez avoir à présent dans le répertoire **Image recognition** les fichiers **yolov3.weights**, **detection.py** et **requirements.txt**

Enfin, il y'a à disposition des images(**.png**) afin de tester l'efficacité de notre programme.

Une fois dans le dossier **Image recognition**, rien de plus simple pour étiqueter notre image que d'exécuter une commande de la forme suivante :

**python detectionv3.py imnames.txt**

Cette image doit se trouver dans le répertoire **Image recognition**

## DEEPFACE

Une fois dans le dossier **DeepFace**, rien de plus simple pour analyser nos visages que d'exécuter une commande de la forme suivante :

**python deepfacev2.py imnames.txt**

Avec imnames.txt un fichier qui contient le nom des images à analyser (un nom par ligne) et j'utilise une méthode **annotate** comme demandé. **annotate** utilise elle-même pour chacune des lignes de **imnames.txt** une méthode **annotate\_image** qui prend en paramètre le nom d'une image !

```
imnames= sys.argv[1]
annotate(imnames)
```

Je produis en sortie un fichier texte qui affiche le nom de chaque image dans l'ordre de lecture puis le contenu de ces images.

(Nom de l'image, label\_name, score et coordonnées du rectangle qui encadre l'image détectée coordonnées : left, right, bottom, top)

```
x = {"Name":image_name,
     "label_name": obj["dominant_race"]+" " +obj["gender"],
     "score": confidence_score,
     "bounding_box":box_coords
    }
```

Après quelques secondes en moyenne pour la prédiction de notre réseau à partir d'une image, vous aurez selon le programme utilisé dans le dossier **DeepFace** ou **Image Recognition** la création d'un fichier nommé "**DataDeepFace.txt**" ou "**DataYolov3.txt**" qui récapitulera les informations demandées en format **JSON** comme demandé dans la consigne.

Exemple: **DataDeepFace.txt**

 DataDeepFace - Bloc-notes

Fichier Edition Format Affichage Aide

```
{"Name": "jackma.jpg", "label_name": "asian Man", "score": "100.0 %", "bounding_box": [108.0, 191.0, 172.0, 42.0]}
{"Name": "obama.jpg", "label_name": "black Man", "score": "100.0 %", "bounding_box": [112.0, 182.0, 102.0, 19.0]}
{"Name": "trump1.jpg", "label_name": "white Man", "score": "94.24 %", "bounding_box": [126.0, 157.0, 152.0, 94.0]}
{"Name": "hp.jpg", "label_name": "white Man", "score": "99.95 %", "bounding_box": [37.0, 81.0, 191.0, 96.0]}
{"Name": "hp.jpg", "label_name": "white Man", "score": "99.92 %", "bounding_box": [212.0, 258.0, 188.0, 86.0]}
{"Name": "hp.jpg", "label_name": "white Man", "score": "95.25 %", "bounding_box": [106.0, 155.0, 178.0, 61.0]}
{"Name": "anjolie.jpg", "label_name": "white Woman", "score": "100.0 %", "bounding_box": [115.0, 182.0, 202.0, 28.0]}
```

Pour supprimer les fichiers créés, rien de plus simple que d'exécuter la ligne suivante.

**rm -f Data\*** dans les 2 répertoires

Je me suis focalisé sur le **genre**(homme/femme) et **l'origine ethnique** des personnes pour la reconnaissance des visages (white/black/asian/...).

Il était aussi possible de travailler sur l'âge et l'émotion des personnes mais cela ne fonctionnait pas parfaitement bien avec **DeepFace**.

Plus d'informations sur le programme Yolov3 sont présentes dans le précédent rapport.

**DeepFace est très performant pour la reconnaissance des visages.**

**Merci de votre attention !**